Deep Learning is Ubiquitous

**Computer Vision**
- Pedestrian and traffic sign detection
- Landmark identification
- Scene recognition
- Medical diagnosis and drug discovery

**Text and Signal Processing**
- Speech Recognition
- Speech & Text Translation

**Robotics & Controls**

MATLAB EXPO 2016

and many more…
What is Deep Learning?

Deep learning performs **end-end learning** by learning **features, representations and tasks** directly from **images, text and sound**

Traditional Machine Learning

- **Manual Feature Extraction**
- **Classification**

<table>
<thead>
<tr>
<th>Car</th>
<th>Truck</th>
<th>Bicycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Deep Learning approach

- **Convolutional Neural Network (CNN)**
- **End-to-end learning**
- **Feature learning + Classification**

<table>
<thead>
<tr>
<th>Car</th>
<th>Truck</th>
<th>Bicycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Demo: Live Object Recognition with Webcam
Why is Deep Learning so Popular?

- **Results:** Achieved substantially better results on ImageNet large scale recognition challenge
  - 95% + accuracy on ImageNet 1000 class challenge

- **Computing Power:** GPU’s and advances to processor technologies have enabled us to train networks on massive sets of data.

- **Data:** Availability of storage and access to large sets of labeled data
  - E.g. ImageNet, PASCAL VoC, Kaggle

---

<table>
<thead>
<tr>
<th>Year</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-2012 (traditional computer vision and machine learning techniques)</td>
<td>&gt; 25%</td>
</tr>
<tr>
<td>2012 (Deep Learning)</td>
<td>~ 15%</td>
</tr>
<tr>
<td>2015 (Deep Learning)</td>
<td>&lt;5%</td>
</tr>
</tbody>
</table>
Two Approaches for Deep Learning

1. Train a Deep Neural Network from Scratch

Lots of data

Convolutional Neural Network (CNN)

Learned features

[95% 3% 2%]

Car ✓
Truck ×
Bicycle ×

2. Fine-tune a pre-trained model (transfer learning)

Medium amounts of data

Fine-tune network weights

Car ✓
Truck ×

MATLAB EXPO 2016
Two Deep Learning Approaches

Approach 1: Train a Deep Neural Network from Scratch

Convolutional Neural Network (CNN)

Learned features

| Car | ✔ |
| Truck | ✗ |
| Bicycle | ✗ |

Recommended only when:

<table>
<thead>
<tr>
<th>Training data</th>
<th>1000s to millions of labeled images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation</td>
<td>Compute intensive (requires GPU)</td>
</tr>
<tr>
<td>Training Time</td>
<td>Days to Weeks for real problems</td>
</tr>
<tr>
<td>Model accuracy</td>
<td>High (can over fit to small datasets)</td>
</tr>
</tbody>
</table>
Two Deep Learning Approaches

**Approach 2:** Fine-tune a pre-trained model (transfer learning)

CNN trained on massive sets of data
- Learned robust representations of images from larger data set
- Can be fine-tuned for use with *new data or task* with small – medium size datasets

![Diagram of fine-tuning network weights](image)

**Recommended when:**

<table>
<thead>
<tr>
<th>Training data</th>
<th>100s to 1000s of labeled images (small)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation</td>
<td>Moderate computation (GPU optional)</td>
</tr>
<tr>
<td>Training Time</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Model accuracy</td>
<td>Good, depends on the pre-trained CNN model</td>
</tr>
</tbody>
</table>
Convolutional Neural Networks

- Train “deep” neural networks on structured data (e.g. images, signals, text)
- Implements Feature Learning: Eliminates need for “hand crafted” features
- Trained using GPUs for performance
Convolutional Neural Networks

Input Image

Convolution → RELU → Pooling → Convolution → RELU → Pooling → Convolution → RELU → Pooling → Convolution → RELU → Pooling → Fully Connected layers to support classification → Flower, Cup, Car, Tree

Sliding window

Filters: light and dark

Simple shapes → complex shapes

Every feature map output is the result of applying a filter to the image. The new feature map is the next input.

Activations of the network at a particular layer

MATLAB EXPO 2016
Convolution Layer

- Core building block of a CNN
- Convolve the filters sliding them across the input, computing the dot product

- Intuition: learn filters that activate when they “see” some specific feature
Rectified Linear Unit (ReLU) Layer

- Frequently used in combination with Convolution layers
- Do not add complexity to the network
- Most popular choice: $f(x) = \max(0, x)$, activation is thresholded at 0
Pooling Layer

- Perform a **downsampling** operation across the spatial dimensions
- Goal: progressively decrease the size of the layers
- Max pooling and average pooling methods
- Popular choice: Max pooling with 2x2 filters, Stride = 2
# Challenges using Deep Learning for Computer Vision

<table>
<thead>
<tr>
<th>Steps</th>
<th>Challenge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importing Data</td>
<td>Managing large sets of labeled images</td>
</tr>
<tr>
<td>Preprocessing</td>
<td>Resizing, Data augmentation</td>
</tr>
<tr>
<td>Choosing an architecture</td>
<td>Background in neural networks (deep learning)</td>
</tr>
<tr>
<td>Training and Classification</td>
<td>Computation intensive task (requires GPU)</td>
</tr>
<tr>
<td>Iterative design</td>
<td></td>
</tr>
</tbody>
</table>
Demo: Classifying the CIFAR-10 dataset

Objective: Train a Convolutional Neural Network to classify the CIFAR-10 dataset

Data:

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Thousands of images of 10 different Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response</td>
<td>AIRPLANE, AUTOMOBILE, BIRD, CAT, DEER, DOG, FROG, HORSE, SHIP, TRUCK</td>
</tr>
</tbody>
</table>

Approach:
- Import the data
- Define an architecture
- Train and test the CNN

Data Credit: Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.
https://www.cs.toronto.edu/~kriz/cifar.html
Demo: Classifying the CIFAR-10 dataset

```matlab
%% Download the CIFAR-10 dataset
if ~exist('cifar-10-batches-mat', 'dir')
    cifar10Dataset = 'cifar-10-matlab';
    disp('Downloading 174MB CIFAR-10 dataset...');
    websave(fullfile(cifar10Dataset, 'tar.gz'))
end
```
Addressing Challenges in Deep Learning for Computer Vision

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing large sets of labeled images</td>
<td><code>imageSet</code> or <code>imageDataStore</code> to handle large sets of images</td>
</tr>
<tr>
<td>Resizing, Data augmentation</td>
<td><code>imresize</code>, <code>imcrop</code>, <code>imadjust</code>, <code>imageInputLayer</code>, etc.</td>
</tr>
<tr>
<td>Background in neural networks (deep learning)</td>
<td>Intuitive interfaces, well-documented architectures and examples</td>
</tr>
<tr>
<td>Computation intensive task (requires GPU)</td>
<td>Training supported on GPUs</td>
</tr>
<tr>
<td></td>
<td>No GPU expertise is required</td>
</tr>
<tr>
<td></td>
<td>Automate. Offload computations to a cluster and test multiple architectures</td>
</tr>
</tbody>
</table>
Demo

Fine-tune a pre-trained model (transfer learning)

Pre-trained CNN (AlexNet – 1000 Classes)

SUV

Car

New Task – 2 Class Classification

New Data
Demo

Fine-tune a pre-trained model (transfer learning)

%% Fine Tuning A Deep Neural Network
1 % This example shows how to fine tune a pre-trained deep convolutional neural network (CNN) for a new recognition task.
2
3 %% Load network
4 cnnMatFile = fullfile(pwd, 'imagenet-cnn.mat');
5 if ~exist(cnnMatFile,'file')
6 disp('Run downloadAndPrepareCNN.m to download and prepare the network.')
7 end

MATLAB EXPO 2016
## Addressing Challenges in Deep Learning for Computer Vision

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing large sets of labeled images</td>
<td><code>imageSet</code> or <code>imageDataStore</code> to handle large sets of images</td>
</tr>
<tr>
<td>Resizing, Data augmentation</td>
<td><code>imresize</code>, <code>imcrop</code>, <code>imadjust</code>, <code>imageInputLayer</code>, etc.</td>
</tr>
<tr>
<td>Background in neural networks (deep learning)</td>
<td>Intuitive interfaces, well-documented architectures and examples</td>
</tr>
<tr>
<td>Computation intensive task (requires GPU)</td>
<td>Training supported on GPUs, No GPU expertise is required</td>
</tr>
<tr>
<td>Automate. Offload computations to a cluster and test multiple architectures</td>
<td></td>
</tr>
</tbody>
</table>
Key Takeaways

- Consider Deep Learning when:
  - Accuracy of traditional classifiers is not sufficient
    - *ImageNet classification problem*
  - You have a pre-trained network that can be fine-tuned
  - Too many image categories (100s – 1000s or more)
    - *Face recognition*